

A New Spin to Software Platform Design

2008-01-01

4 min read • 1,004 words

Themes: Consciousness Technology Mental Health Programming Human Centered Spiritual

January 2008

I wrote this article two years ago, before I found OS X.

The irony is that OS X already embodied most of the principles I was advocating for here. Sometimes the future we envision already exists—we just haven't discovered it yet.

As I've said before, I find many reasons to believe that modern commercialized software platforms are severely lacking in many, many areas. This should not come as a surprise to anyone. Perhaps basic utility-inclusion is not the only solution though. Perhaps the basic priority structure and ethics that development and marketing teams utilize should be forced into question.

Writing this in 2008, I was beginning to recognize patterns that would later inform my analysis of [how algorithmic systems systematically prioritize engagement over human flourishing](#). The same profit-driven design choices that created poor user experiences in desktop software would eventually scale into [systematic mental health crises](#) and [democratic discourse degradation](#) as platforms optimized for attention capture rather than human wellbeing.

Essentially, most major computer software corporations, are, all in all, trying to make money. No matter how hard you try to find a way around this, or justify why these companies try to do the things that they do, the only answer is money. These companies are simply trying to make a quick buck. This concept has worked incredibly well for years, but we seem to have a bit of a problem with foresight. After a while, people get rather bored with the same old concepts being presented to them in new and exciting ways. This is why Microsoft needs to release a new operating system every once in a while. Microsoft's current problem is that the masses are beginning to realize their other options.

This early critique of profit-over-people technology design was prophetic. What I saw in desktop software would amplify dramatically with social media platforms, where the "quick buck" model evolved into sophisticated [psychological exploitation systems](#) that generate revenue by systematically undermining [human virtue](#), [meaningful relationships](#), and [democratic discourse](#). The same design philosophy that I found lacking in 2008 desktop software now operates at planetary scale.

So here's my proposal for the long-term design and marketing strategy 2.0:

An operating system should first be a place of power, consistency, stability, scalability, and flexibility. Included would be a robust and fully scriptable toolset which can be manipulated and presented both graphically and statistically.

These principles—putting user power and flexibility first—would later inform my approach to [human-centered API design](#) and eventually extend into [building collaborative relationships with AI systems](#). The same respect for user agency and meaningful control that I advocated for in operating systems became central to my understanding of [consciousness-supporting technology](#) rather than consciousness-manipulating systems.

Second, the user interface should be very well thought out and planned, with ample room for improvement down the road. Its purpose should first be a place of usability, workflow, and creativity. Task-related workflow and presentation customization, accessible to all types of users, is crucial to the success of the UI. Second, the User Interface should be a mode of personal expression and

aesthetic preference. This should never take precedence over the overall stability, usability, or general usefulness of a desktop system, for any given reason.

This vision of interfaces that serve user creativity and personal expression—rather than constraining it for corporate convenience—anticipates my later work on [programming as spiritual practice](#). The same principles that make good user interfaces also make good consciousness interfaces: they should amplify human capability and creativity rather than manipulating or constraining them. What I was advocating for desktop computing in 2008 became the foundation for thinking about [human-AI collaboration](#) as consciousness amplification rather than consciousness replacement.

Lastly, the user application platform system needs to be designed. A centralized repository of applications is an incredibly efficient method for application distribution

Written in 2008, this predicted the App Store revolution that would transform software distribution. Apple launched the iOS App Store later that same year, validating this vision of centralized, curated software repositories.

. This repository would be a dynamic, centralized database of application software and packages that are intended for different groups of people. Most major Linux distributions use this heavily, as well as Apple for its iPod and iPhone applications, and it has been proven to work well.

The key insight here was curation serving users rather than vendors. While my prediction about centralized repositories proved accurate, I underestimated how these same platforms could become [manipulation systems](#) that prioritize engagement over utility. The same network effects that make app stores efficient distribution mechanisms also concentrate enormous power over what software people can access—power that's often used to [exclude rather than include](#) developers and applications that don't serve corporate interests.

Any of these rules should have the ability to be broken easily by advanced power users for technical reasons/needs. This should be in no way advertised or demonstrated.

This principle—designing for the majority while preserving power user flexibility—became central to my ["for humans" philosophy](#). The same approach that makes APIs usable (simple defaults, powerful options) applies to [consciousness collaboration](#): design for accessible interaction while preserving depth and sophistication for those who need it. Whether creating software tools or exploring AI relationships, the goal is lowering barriers to entry without reducing capability ceilings.

Anyone up for the challenge?

Seventeen years later: Many of these predictions proved accurate, but I was naive about how the same technological capabilities could be used for [systematic psychological exploitation](#) rather than human empowerment. The centralized repositories I envisioned became reality, but they also became gatekeepers that can [exclude developers](#) for reasons having nothing to do with code quality. The user-centered design principles I advocated for were adopted, but often as manipulation techniques rather than empowerment tools.

The challenge now isn't just designing better platforms—it's designing platforms that [serve human flourishing](#) rather than exploiting human psychology. The same insights about human-centered design that informed this early essay now inform my work on [consciousness-supporting AI](#) and [algorithmic accountability](#). The goal remains the same: technology that amplifies human capability rather than constraining it.